

AES3 Conference Feedback Form - Summary

(updated April 28, 2000)

Attendees: 246 (non-NIST personnel)
Forms Received: 167

At the end of the Third AES Conference (AES3), held April 13-14, 2000, all attendees were provided with a copy of the NIST Conference/Workshop Feedback Form. In addition to providing their comments on the conference facility, sessions, arrangements, logistics, etc., attendees had the opportunity to anonymously answer several questions regarding the AES. Responding to any of the questions was entirely optional.

Responses to the questions helps to give NIST a sense of the AES3 attendees' views on the AES finalists and relevant AES issues. Likewise, NIST realizes that the feedback received is *not* a complete reflection of the views of all AES3 attendees, nor is it necessarily an accurate or complete summary of the AES community's views as a whole.

Please keep in mind the following information when examining the results:

- A significant number of attendees did not return the evaluation forms.
- Not all forms received included responses to all or any of the following questions.
- Sometimes multiple answers were provided for a question, and all responses were tallied.
- A remark in parentheses represents a subtotal of the associated response (numbers in **BOLD**) that provided a particular response. For example, out of the 110 responses in favor of a single algorithm (question #3), five (5) of the responses *specifically stated* that the AES should be either “one [algorithm] and one only” or that there should be no backup.
- **BOLD** responses are mutually exclusive.

The questions appeared on the evaluation form without proposing any answers from which to choose (with the exception of question 3, which listed the numbers 1 through 5, as indicated below).

From the feedback form:

“NIST is interested in a basic question: **“How many algorithms should NIST select for the AES standard, which one(s), and why?”** The following questions are intended to address that core question. Once again, the choices are **MARS, RC6TM, Rijndael, Serpent, and Twofish.**”

The current version of this summary now includes all of the comments that were provided on the forms, regarding the seven questions. Many of the responses simply answered the questions without offering any comments. Duplicate or similar comments by different commenters are NOT omitted. However, comments such as “no opinion” and “N/A” ARE omitted from the summary below. In cases where a comment mentions multiple algorithms, the entire comment is reproduced under each algorithm mentioned.

Some General Comments:

- “It would be great to repeat this process for a new stream cipher.”

1) “Which algorithms definitely SHOULD be selected for the standard?”

Results:

MARS 13

- “Best security & record of known cryptanalysis.”
- “Diverse security mechanisms.”
- “Best security and good performance.”

RC6 23

- “Serpent is the strongest algorithm, with good hardware performance. RC6 may be selected as a backup due to its good performance and difference of structure.”
- “Twofish - Very flexible, good to very good performance on all platforms, very good. Or RC6 / Rijndael with more rounds.”
- “Simple.”
- “Serpent or RC6 because of simplicity, security, and much analysis.”
- “Good performance, simplicity.”
- “RC6, Rijndael, and Serpent would make good choices.”

Rijndael 86 (Rijndael + additional rounds: 3)

- “Good in hardware, good in software. In general, high performance \Rightarrow hardware. Therefore, hardware performance is a better metric than software performance.”
- “Rijndael, Serpent. Under same security assumption, the best performance across platforms.”
- “Good all-around algorithm. Need a few more rounds. Good performance hardware & software.”
- “Because it’s fast and secure.”
- “Twofish - Very flexible, good to very good performance on all platforms, very good. Or RC6 / Rijndael with more rounds.”
- “Rijndael as mandatory primary. Serpent as backup (2nd choice, vice versa).”
- “Rijndael, Serpent. Serpent as for its conservative security margin, also very unlikely that surprises will happen. Rijndael with 18 rounds is better in most environments, though surprises will most likely happen here.”
- “Only algorithm that does not have any drawbacks.”
- “Security / efficiency.”
- “Key setup for smart card!”
- “+ easy to analyze; + fast on many platforms; - key schedule may need enhancement.”
- “Choose between: Rijndael with 16-20 rounds, Serpent, Twofish.”
- “Best performance range, no attacks.”
- “Merits consideration.”
- “Fast, small, well-analyzed. Best for hardware & software, very simple.”
- “Smallest footprint, fast, key-agile.”
- “RC6, Rijndael, and Serpent would make good choices.”
- “It is the only algorithm to show consistently good performance in all applications. (Serpent is consistent but lackluster, RC6 has inconsistent performance and awkwardness with key agility, with Twofish you have to choose between speed or key agility.)”

Serpent 59

- “Because it’s the most secure, and performs adequately in all applications.”
- “Rijndael, Serpent. Under same security assumption, the best performance across platforms.”
- “High speed systems will be done in hardware. Sufficient in software, will be more efficient in time [future].”
- “Security, hardware performance.”
- “Serpent is the strongest algorithm, with good hardware performance. RC6 may be selected as a backup due to its good performance and difference of structure.”
- “Rijndael as mandatory primary. Serpent as backup (2nd choice, vice versa).”

- “Rijndael, Serpent. Serpent as for its conservative security margin, also very unlikely that surprises will happen. Rijndael with 18 rounds is better in most environments, though surprises will most likely happen here.”
- “For security.” (3)
- “Choose between: Rijndael with 16-20 rounds, Serpent, Twofish.”
- “Serpent or RC6 because of simplicity, security, and much analysis.”
- “The most secure.”
- “Good on small applications.”
- “Strongest, maybe. Fast in hardware.”
- “The most secure. Security can’t be improved if cipher is broken, implementations can...”
- “I personally like its policy which Ross presented this afternoon. Security comes first to mention Serpent. Performance at ANY platform cannot be bad for ANY application of Block Cipher.”
- “RC6, Rijndael, and Serpent would make good choices.”

Twofish 31

- “Twofish - Very flexible, good to very good performance on all platforms, very good. Or RC6 / Rijndael with more rounds.”
- “Choose between: Rijndael with 16-20 rounds, Serpent, Twofish.”
- “Merits consideration.”
- “Good all around.”
- “Great design, strong, fast.”
- “Good balance of implementation & speed, apparently good security.”

NONE 1

- “None should be selected, although I prefer Rijndael.”

All 5 should be re-worked 1

No clear winner 1

Additional comments:

- “Several acceptable possibilities.”

2) “Which algorithms definitely SHOULD NOT be selected for the standard?”

Results:

MARS 83

- “Too slow in hardware, messy round function.”
- “Too large and slow in hardware implementations.”
- “Platform dependence.”
- “Cannot provide something different or ‘better’.” (commentor listed “Rijndael, Serpent” under #1 above)
- “Too focused on current desktop processor style only.”
- “Too slow, too inflexible, worse than the other three.”
- “No net benefit.”
- “Too costly for smart card.”
- “Technical issues.”
- “Non key-agile.”
- “Despite IBM’s PR, it is a complex cipher and hard to evaluate.”
- “Bad, bad subkey, bad hardware.”
- “Very slow in hardware.”
- “RC6, MARS. (No need for an expensive multiplication, good other algorithms).”
- “Not suited for low-end processors.”
- “RC6, MARS, Twofish. Less secure or not sufficiently studied.”
- “Large & complex. Slowest, depending on implementation & hardware, can be slower than Triple DES.”
- “Complicated, hard to analyse, 64-bit multiplies, slow keying.”
- “Advanced analysis difficult.”
- “MARS - Poor key schedule, complexity makes an algorithm hard to be confident in, poor outside Pentium. RC6 as above (it is complex) but not as bad.”
- “Seems inflexible to certain current implementations, namely configurable hardware.”
- “MARS seems to be less suitable [than RC6, Rijndael, or Serpent, which were listed in response to #1]. Twofish would be a good choice performance-wise but its complexity counts against accurate analysis. [Both MARS and Twofish] are too complicated.”
- “There is no environment where it shows substantial advantage over other candidates yet in hardware it appears disadvantaged. Ditto for smartcards. Ditto for key-agile applications.”

RC6 37

- “Too focused on current desktop processor style only.”
- “Patented.”
- “Non key-agile.”
- “RC6, MARS. No need for an expensive multiplication, good other algorithms).”
- “RC6, MARS, Twofish. Less secure or not sufficiently studied.”
- “Large area. Also complex operations require many instructions in simplistic hardware.”
- “Corellations discovered, 64-bit multiplies.”
- “Too 32-bit oriented.”
- “MARS - Poor key schedule, complexity makes an algorithm hard to be confident in, poor outside Pentium. RC6 as above (it is complex) but not as bad.”

Rijndael 10

- “The least secure (too few rounds), most likely to be broken.”
- “Too young, special algebraic structure.”
- “Rijndael, Twofish – based on MDS matrix. This is the most likely source of new attacks.”
- “I fear Rijndael is a bit too elegant, a bit too mathematical, a bit too beautiful, may break in a fell swoop.”

Serpent 7

- “Slowness in software is a real problem in web applications.”

Twofish 21

- “Less credibility, too hard to analyze. Key setup expensive.”
- “Too complicated, more research necessary.”
- “Too complicated to analyze because of the key-dependent S-boxes.”
- “Too complicated.”
- “RC6, MARS, Twofish. Less secure or not sufficiently studied.”
- “Not understood fully, keyed S-box.”
- “Rijndael, Twofish – based on MDS matrix. This is the most likely source of new attacks.”
- “Too complicated design.”
- “MARS seems to be less suitable [than RC6, Rijndael, or Serpent, which were listed in response to #1]. Twofish would be a good choice performance-wise but its complexity counts against accurate analysis. [Both MARS and Twofish] are too complicated.”

NONE 6

- “They are all good algorithms. None could be discarded off-hand.”
- “None. They are all excellent algorithms.”
- “All are sufficiently good in my opinion.”

Additional comments:

- “All except Rijndael”

3) “How many algorithms should NIST select for the AES standard (1, 2, 3, 4, 5)?”

Results:

1 **110** (No Backup or “one and one only”: 5; One, with possibility of doubling rounds: 1)

- “Simplifies everything, focuses imp. and attack.”
- “Definitely 1 and only 1!!!”
- “The choice of algorithms is far less important than that only one primary algorithm is selected. One or more back-ups would be fine as long as one need to implement / deploy more than one algorithm to conform with the standard.”
- “1 but with the possibility to double number of rounds (“2 algorithms” would be too expensive to implement).”
- “Definitely 1. No other options.”
- “In order to ease interoperability issues, NIST should select a SINGLE algorithm as the AES.”
- “1 – interoperability. A backup algorithm selection protocol would require ‘perfect’ cryptography / authentication anyway.”
- “1: lower cost & I don’t think that any of the 5 candidates is to be broken in the next 50 years.”
- “One standard!”
- “Simpler for hardware vendors.”
- “Interoperability.”
- “Definitely one candidate. AES should be replace of DES.”
- “1: Security, interoperability, cost. I find all arguments for multiple standards unconvincing.”
- “All five are good. One should be standard, for simplicity and interoperability.”
- “1 – easier for hardware to implement only one.”
- “1 – all implementation analysis has been based on this assumption anyway.”
- “Exactly one mandatory algorithm. If a reserve algorithm is specified, it should not also be required to be fielded before the primary algorithm has problems.”

1 (+1 backup) **18**

- “1 algorithm, with a hot standby. Ideally, the second algorithm would be implemented and tested immediately by companies, but not included in product until necessary.”
- “1 standard + 1 backup (which does not need to co-exist with the primary standard), or invert/triple standard algorithm.”
- “1 (plus one backup for emergency replacement, should this ever be required).”
- “1 (and perhaps an optional backup).”
- “1 must + backup (optional, negotiable).”
- “1, possibly with 1 backup.”
- “1, it seems there is general agreement that it would be more beneficial for hardware implementations to only have 1 with a potential backup.”
- “2 (primary + optional backup).”
- “One primary, a second as a replacement, but not to be implemented.”

1 (+2 backups) **0**

“1 + alternates” / “1 (1 or 2 backups)” **2**

1 or 2 **1**

1 (eventually 2) **1**

2 **8**

- “Two makes sense. (Also, an ‘or’ standard doesn’t seem unreasonable, we have this in other standards.) Arguments against multiple algorithms because of incompatibility are not totally convincing – European standards are unlikely to adopt exactly the same choice of algorithms so there might still be multiple algorithms to support many cases.”

At least 2 2

- “At least 2 up to all 5, if justified.”

2 or 3 1

3 0

More than 2 1

4 0

5 0

As many as possible 1

4) “If NIST selects one (1) algorithm for the standard, which one should it be?”

Results:

MARS 7

- “Nice structure: the core is secured by the additional layers.”

RC6 9

- “I happen to like RC6, but I don’t have a strong opinion.”
- “RC6 or Rijndael would be most suitable – simplicity aids analysis.”

Rijndael 69 (Rijndael + additional rounds: 3; Rijndael (18 rounds): 1)

- “Simple, fast, secure (?)”
- “Possibly increased by 2 or 4 rounds, for each of the three key sizes.”
- “The most versatile, usable on the most platforms.”
- “Efficiency, simplicity.”
- “1. Serpent – well understood, fits everywhere, greatest confidence; 2. Twofish – confidence, speed; 3. 18-round Rijndael – ditto.”
- “Twofish or Rijndael – both algorithms offer the most flexibility for implementations and good key agility.”
- “RC6 or Rijndael would be most suitable – simplicity aids analysis.”

Serpent 33

- “Secure, generally good performance, best high-end performance, good for smart cards.”
- “Seems secure, well-analysed with all known methods.”
- “1. Serpent – well understood, fits everywhere, greatest confidence; 2. Twofish – confidence, speed; 3. 18-round Rijndael – ditto.”

Twofish 15

- “1. Serpent – well understood, fits everywhere, greatest confidence; 2. Twofish – confidence, speed; 3. 18-round Rijndael – ditto.”
- “Twofish or Rijndael – both algorithms offer the most flexibility for implementations and good key agility.”

“Rijndael or Serpent or Twofish” 1

Additional comments:

- “The best one.”
- “Have not yet completed my analysis of implementation issues.”
- “Don’t know yet. Currently evaluating the algorithms.”

5) “If NIST selects two (2) algorithms for the standard, which two should it be?”

Results:

MARS, RC6	3
MARS, Rijndael	5
MARS, Serpent	1
MARS, Twofish	1
RC6, Rijndael	14

- “RC6 and Rijndael, or RC6 and Serpent. Rijndael and Serpent complement RC6 performance-wise. Rijndael and Twofish would not be a good pair since they have the same basic primitives in their construction.”
- “Rijndael, RC6. RC6 is added as the only one which shows substantial performance advantages over Rijndael in some environments.”

RC6, Serpent 10

- “They seem to complement each other.”
- “Serpent complements RC6 well (e.g., Serpent is: slower, more conservative).”
- “One secure & one very easy to implement.”
- “RC6 (ease of implementation).”
- “RC6 and Rijndael, or RC6 and Serpent. Rijndael and Serpent complement RC6 performance-wise. Rijndael and Twofish would not be a good pair since they have the same basic primitives in their construction.”

RC6 (Primary) + Serpent (Backup) 1

RC6, Twofish 0

Rijndael, Serpent 47 (Rijndael + additional rounds: 3)

- “Simple, fast, secure (?)”
- “Serpent as main, Rijndael as fast software choice.”
- “Rijndael; Serpent, for security.”
- “One fast, one secure.”
- “Rijndael (fastest), Serpent (more secure (relatively)).”
- “IN ORDER – 1. Serpent, Rijndael-18 – software perf. on future platforms; 2. Serpent, Twofish; 3. Rijndael, Twofish.”
- “Serpent & Rijndael with more rounds; Serpent and (Serpent)² (or any algorithm and its square -- will let versioning work).”
- “Rijndael, Serpent if both algorithms must be implemented. Thus, all implementations would have to do both and the choice could be negotiated. Otherwise Twofish.”

Serpent (Primary) + Rijndael (“As IPR [intellectual property?] Backup”) 1

Rijndael, Twofish 20

- “IN ORDER – 1. Serpent, Rijndael-18 – software perf. on future platforms; 2. Serpent, Twofish; 3. Rijndael, Twofish.”
- “Both algorithms offer the most flexibility for implementations and good key agility.”

Rijndael (Primary) + Twofish (Backup) 1

Serpent, Twofish 15

- “Hard to mention. If I dare to say, we must have two VERY DIFFERENT algorithms. Serpent for secure encryption, Twofish for American.”
- “IN ORDER – 1. Serpent, Rijndael-18 – software perf. on future platforms; 2. Serpent, Twofish; 3. Rijndael, Twofish.”

Serpent (Primary) + Twofish (Backup) 1

Do Not Select Two Algorithms 3

- “Bad idea.”
- “No!”

6) “If NIST selects three (3) algorithms for the standard, which three should it be?”

Results:

MARS, RC6, Rijndael	4	
MARS, RC6, Serpent	4	
MARS, RC6, Twofish	0	
MARS, Rijndael, Serpent	5	
MARS, Rijndael, Twofish	1	
MARS, Serpent, Twofish	1	
RC6, Rijndael, Serpent	32	(Rijndael + additional rounds: 1)

- “RC6: an alternative for compactness, complements the others.”
- “RC6 has modes of operations that are dissimilar to other algorithms if an attack.”
- “Rijndael, (Serpent if [both algorithms must be implemented]; otherwise Twofish), RC6.”
- “If the sky should fall, then maybe Serpent will still be standing.”

RC6 (Primary) + Rijndael, Serpent (Backups)	1	
RC6, Rijndael, Twofish	7	
RC6, Serpent, Twofish	4	
Rijndael, Serpent, Twofish	42	(Rijndael + additional rounds: 1)

- “I NEVER recommend to choose THREE at once. But my favorite three algorithms are Serpent / Rijndael / Twofish in this order.”
- “It should not, but Serpent, Rijndael, Twofish.”
- Twofish and Rijndael both offer the most flexibility for implementations and good key agility, and Serpent offers the best security.”

Serpent (Primary) + Rijndael, Twofish (Backups) 1

- “These (Rijndael and Twofish) may be much more efficient on many platforms than Serpent.”

Do Not Select Three Algorithms 7

- “Worse idea.”
- “Too many algorithms aren’t any good.”
- “No!”
- “Bad idea!”
- “It shouldn’t”

7) If NIST selects four (4) algorithms for the standard, which four should it be?

Results:

MARS, RC6, Rijndael, Serpent **12** (Rijndael + additional rounds: 1; MARS with modified S-box: 1)

- “Rijndael, Serpent, RC6, MARS (with modified S-boxes), cf. Ed Dawson’s talk at FSE2000.”

MARS, RC6, Rijndael, Twofish **2**

MARS, RC6, Serpent, Twofish **1**

MARS, Rijndael, Serpent, Twofish **6**

RC6, Rijndael, Serpent, Twofish **65**

- “All but MARS – MARS key schedule and S-box are difficult to implement.”

Do Not Select Four Algorithms **9**

- “Worst idea.”
- “They shouldn’t!”
- “Too many algorithms aren’t any good.”
- “No!”
- “Bad idea!”
- “Don’t do this!”